Arduino 101 & 102
Workbook & Reference Material

Earl Girbovan
Seth Neumann

Contents

Blink an LED

```
/* Blink

This program blinks an LED on and off
In the TOOLS BOARD tab in the Arduino IDE, ensure 'Arduino Nano w/ATmega328' is selected

*/

int LED2 = 2;  //D2

void setup()
{
  pinMode( LED2, OUTPUT );

}

void loop()
{
  digitalWrite( LED2, HIGH );
  delay( 1000);
  digitalWrite( LED2, LOW );
  delay ( 1000 );

}
```

Fire Effect Code

```
/* LED Fire Effect
   Uses two amber and one red LED to simulate fire
   From the instructables.com web site, slightly modified
*/

int ledPin1 =  9;
int ledPin2 = 10;
int ledPin3 = 11;

void setup()
{
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
}

void loop()
{
analogWrite(ledPin1, random(135, 255));
analogWrite(ledPin2, random(135, 255));
analogWrite(ledPin3, random(135, 255));
delay(random(1, 100));
}
```

While Loop: Turning the fire off and on

```
/* Intermittent LED Fire Effect
  Uses two amber and one red LED to simulate fire
  Included a WHILE loop, so that the effect turns on and off.
*/

int ledPin1 =  9;
int ledPin2 = 10;
int ledPin3 = 11;
int OnTime;
int OffTime;
int i;

void setup()
{
        pinMode(ledPin1, OUTPUT);
        pinMode(ledPin2, OUTPUT);
        pinMode(ledPin3, OUTPUT);
}

void loop()
{
        OnTime = random( 1000, 5000 );
        while( OnTime > 0 )
                {
                analogWrite(ledPin1, random(120)+135);
                analogWrite(ledPin2, random(120)+135);
                analogWrite(ledPin3, random(120)+135);
                i = random(1, 100);
                delay( i );
                OnTime = OnTime - i;
                }

         // off time, ensure the LEDs are turned off
         analogWrite(ledPin1, 255 );
         analogWrite(ledPin2, 255 );
         analogWrite(ledPin3, 255 );
         OffTime = random ( 1000, 3000 );
         delay ( OffTime );
}
```

Servo Sweep

```
/*
Servo Sweep

This program slowly sweeps a servo back and forth from 0 to a presettable maximum number
of degrees.
*/

#include <Servo.h>

Servo Servo1;

int pos;
int MaxPos = 60;
int DelayTime = 90;

void setup()
{
 //tell the Arduino & the servo driver that the servo is connected to pin 7
 Servo1.attach(7);
}

void loop()
{
 //sweep the servo back and forth
 for( pos = 1; pos <=MaxPos; pos++ ) //sweep from 0 to MaxPos in 1 degree steps
 {
  Servo1.write(pos);
  delay(DelayTime);
 }
 delay( 1000 );

 //now sweep it back
 for( pos = MaxPos; pos >=1; pos--)  //sweep from MaxPos to 0 in 1 degree steps
 {
  Servo1.write(pos);
  delay(DelayTime);
 }
 delay( 1000 );
}
```

Grade Crossing Simulator

```
/*
Grade Crossing

Ties together the blinking lights and servomotor sweep,
along with reading an input, to drive a grade crossing.
The servo simulates a crossing arm.

The sequence is a bit stylized to keep the code manageable
for the clinic.
*/

#include <Servo.h>
Servo Servo1;

int pos;
int MaxPos   = 90;   // max position in degrees
int ServoPos;        // where the servo is in it's sweep
int StepTime = 90;   // time between steps
int ServoTime;       // time since last step

const int TriggerPin = 12;  //Digitial input D12
int Trigger;

int BlinkRate = 1000;  //rate at which the LED blinks
int LEDTime;
int Delta = 90;    // number of msec for each loop iteration
int LEDPin = 2;
boolean LEDState;

void setup()
{
  pinMode( TriggerPin, INPUT_PULLUP);
  //tell the Arduino & the servo driver that the servo is connected to pin 7
  Servo1.attach(7);
  Servo1.write( 0 );   // ensure the crossing gate is raised
  ServoPos = 0;
  LEDState = false;    // LED starts off
  pinMode ( LEDPin, OUTPUT );
  digitalWrite ( LEDPin, HIGH );
  } // end of setup
```

```
void loop()
{
 //poll, looking for a LOW to start the sequence
 do
 {
  Trigger = digitalRead( TriggerPin );
 }
 while( Trigger == HIGH );

 //start the sequence
 ServoTime  = StepTime;   // initialize the crossing arm timer
 LEDTime    = 0;          // go into the routine with the LED off, so it will turn on
 LEDState   = false;
 ServoPos   = 0;         // crossing arm raised position
 do
  {
  // see if we need to change the stat of the LED
  LEDTime = LEDTime - Delta;
  if( LEDTime <= 0 )
   {
    LEDState = !LEDState;  //LED to other state; on or off
    if (LEDState)
    {
     digitalWrite ( LEDPin, LOW );
    }
    else
    {
     digitalWrite ( LEDPin, HIGH );
    }
    LEDTime = BlinkRate;  //load up the next blink cycle
   }

  //see if we need to change the state of the crossing arm
  ServoTime = ServoTime - Delta;
  if( ServoTime <= 0 )
   {
    //if arm not at down position, move it
    if( ServoPos < MaxPos )
     {
      ServoPos++;
      Servo1.write( ServoPos );
      // and reset the timer for the next step
      ServoTime = StepTime;
     }
```

```
    }
  delay ( Delta );
  Trigger = digitalRead( TriggerPin );
  }
 while ( Trigger == LOW );

 //switch has been released, turn off the light and raise the arm
 digitalWrite( LEDPin, HIGH );

 for( int pos = ServoPos; pos >=1; pos--)
 {
  Servo1.write(pos);
  delay( StepTime );
 }


} // end of loop
```

Servo with Travel Limits Set by Pots

```
#include <Servo.h> // servo Sweep is by BARRAGAN <http://barraganstudio.com> This example code is in the public domain.
Servo myservo;     // create servo object to control a servo
                   // twelve servo objects can be created on most boards

int pos;

//const int LED9 = 9; PWM functionality diabled by using servo library
//const int LED10 = 10; PWM functionality diabled by using servo library

int pot0val = 0; // value for pot read
int pot1val = 0; // value for pot read
int lowpos = 1;  // set it in the range of servo
int highpos = 127;// set it in the range of servo

void setup() {

myservo.attach(7);  // attaches the servo on pin 7 to the servo object

//byte i;
}

void loop() {

 pot0val = analogRead (0); // first pot
 lowpos = pot0val/6; //scale to 1-~180, this is actually 170 degrees
 pot1val = analogRead (1); // second pot
 highpos = pot1val/6; //scale to 1-~180
 {for ( pos=lowpos; pos<=highpos; pos++){

   myservo.write(pos);              // tell servo to go to POSition
   delay(20);                       // waits 15ms for the servo to reach the position
  }
 delay(500); // time to recyle
  }

 // from Banzi example 6B
}
```